Encoding Generative Adversarial Networks for defense against image classification attacks

José M. Pérez-Bravo[1], José A. Rodríguez-Rodríguez[1], Jorge García-González[1,2][0000−0001−8610−3462], Miguel A. Molina-Cabello[1,2][0000−0002−8929−6017], Karl Thurnhofer-Hemsi[1,2][0000−0001−6519−1213], and Ezequiel López-Rubio[1,2][0000−0001−8231−5687]

[1] Department of Computer Languages and Computer Science
University of Málaga, Málaga, Spain
[2] Instituto de Investigación Biomédica de Málaga – IBIMA, Málaga, Spain
{josperbra,joseantoniorodriguez}@uma.es ;
{jorgegarcia,miguelangel,karlkhader,ezeqlr}@lcc.uma.es

Abstract. Image classification has undergone a revolution in recent years due to the high performance of new deep learning models. However, severe security issues may impact the performance of these systems. In particular, adversarial attacks are based on modifying input images in a way that is imperceptible for human vision, so that deep learning image classifiers are deceived. This work proposes a new deep neural network model composed of an encoder and a Generative Adversarial Network (GAN). The former encodes a possibly malformed input image into a latent vector, while the latter generates a reconstructed image from the latent vector. Then the reconstructed image can be reliably classified because our model removes the deleterious effects of the attack. The experiments carried out were designed to test the proposed approach against the Fast Gradient Signed Method attack. The obtained results demonstrate the suitability of our approach in terms of an excellent balance between classification accuracy and computational cost.

Keywords: Generative Adversarial Networks · Adversarial attack · Fast Gradient Signed Method attack.

1 Introduction

Deep learning (DL) has been widely used during the last decade for many different applications due to its exceptional performance. Particularly, Convolutional Neural Networks (CNNs) have become a standard in most image processing tasks, such as detection [1], segmentation [2], classification [3] or quality enhancement [4]. These deep models outperform classical machine learning methods and provide a powerful tool for scientists and entrepreneurs to develop new solutions.

However, there is a security breach in many existing DL models: perturbed input samples that are imperceptible for humans may provoke wrong outputs

by the networks. This tentative is called an adversarial attack. DL models learn non-intuitive features that adversarial attacks are able to exploit by using manipulations of the inputs [5]. Focusing on the image classification domain, adversarial examples, i.e., perturbed samples, are designed intentionally to cause false predictions. Sometimes, these adversarial samples generated to disturb one model can be transferred to another target model, which is used to perform what is called a black-box attack [6]. On the other hand, when the adversary has access to all the parts of the intrinsic model, this is referred to as a white-box attack [7]. This paper intends to provide defense mechanisms for this type of attack.

Developing defense mechanisms is of paramount importance since attacks can affect many real-world applications. For example, an adversary can modify traffic signs to cause accidents in autonomous vehicles [8]. Many defensive methods for detecting adversarial samples and providing a correct classification have been proposed. Thus, roughly, these approaches can be categorized into two types: heuristic defenses and provable defenses. The former is only experimentally validated, while the latter is theoretically proved. Creating heuristic defenses is, somehow, easier than proving the effectiveness of a provable defense. In this paper, we will focus on heuristic methods. Some of the most representative heuristic defenses are:

- Adversarial training: fast gradient sign method [9], projected gradient descent [10], generative adversarial training [11].
- Randomization: random input transformation [12], random noising [13], random feature pruning [14].
- Denoising: conventional input rectification [15], Generative Adversarial Networks (GANs) based input cleansing [16], auto encoder-based input denoising [17].

Most of the incorrect classifications of adversarial examples are due to imperceptible modifications of the pixels of an image. This work intends to propose a defensive algorithm to reduce the effect of adversarial attacks employing the combination of a GAN-based input cleansing method and an autoencoder. GANs were proposed by Goodfellow et al. [9], being a model composed of two networks: a generator that learns a mapping between a latent space and a data distribution and a discriminative network that distinguish the proper data. The idea of our method is the use of an encoder to project the input image onto the latent space and then feed the generative adversarial network. Thus, given an adversarial example, the latent vector generated by the encoder would be associated with a benign image learned by the GAN.

Therefore, the contributions of the paper are: 1) a new methodology of defending against adversarial examples is proposed combining GANs and autoencoders, named as EGAN, 2) a practical framework for image classification is implemented with a simple training procedure. The rest of the paper is organized as follows: in Section 2 is presented the theory of our proposal, Section 3 is devoted for the experimentation, and finally, the conclusions are presented in Section 4.

2 Methodology

In this section, our proposed deep learning model is presented. It is called Encoding Generative Adversarial Networks (EGAN) because it contains a Generative Adversarial Network (GAN) that produces an image from a latent vector, and an encoder that produces a latent vector from an image. The encoder is a feed-forward deep convolutional neural network.

Let us note G the Generative Adversarial Network:

$$\mathbf{X} = G(\mathbf{z}) \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^L$ is a latent vector, and $\mathbf{X} \in \mathbb{R}^{N \times M \times Q}$ is the generated image with N rows, M columns and Q channels. The latent space dimension L is much smaller than the size of the image, $L \ll NMQ$. On the other hand, let E stand for the encoder:

$$\mathbf{z} = E(\mathbf{X}) \tag{2}$$

As seen, the encoder E performs the inverse operation of the Generative Adversarial Network G. Now, let us assume that G has already been trained on some distribution $P(\mathbf{X})$ of images of interest. In our scheme, G is held fixed so that their parameters are not changed during the training of the encoder E.

The encoder is trained by minimizing the loss function $\mathcal{L}$ given by the mean squared error between the generated image $\mathbf{X}$ and its reconstruction $\hat{\mathbf{X}}$ by the encoder:

$$\hat{\mathbf{X}} = G(E(\mathbf{X})) \tag{3}$$

$$\mathcal{L} = \frac{1}{T} \sum_{i=1}^{T} \left\| \mathbf{X}_i - \hat{\mathbf{X}}_i \right\|^2 \tag{4}$$

where $\|\cdot\|$ stands for the Euclidean norm and T is the number of training images $\mathbf{X}_i$. Please note that in (4) it is assumed that the images are flattened prior to the computation of the Euclidean norm.

The training algorithm for the encoder E reads as follows:

1. Draw T random latent vectors $\mathbf{z}_i \in \mathbb{R}^L$, for $i \in \{1, ..., T\}$.
2. Generate the T associated training images with the GAN: $\mathbf{X}_i = G(\mathbf{z}_i)$.
3. Adjust the trainable parameters of the encoder by stochastic gradient descent on the loss function $\mathcal{L}$ (equation 4).
4. If the maximum number of epochs for the training of the encoder has been reached, then halt. Otherwise, go to step 3.

In our experiments, the attack applied to the input images is called Fast Gradient Signed Method (FGSM) which consists in propagating $\nabla_{\tilde{\mathbf{X}}} J(\theta, \tilde{\mathbf{X}}, \mathbf{Y})$, where $\tilde{\mathbf{X}}$ is the input image, $\mathbf{Y}$ is the ground truth label for $\tilde{\mathbf{X}}$, and θ stands for the parameters of the attacked classifier, to yield the adversarial sample $\mathbf{X}$:

$$\mathbf{X} = \tilde{\mathbf{X}} + \epsilon * sign(\nabla_{\tilde{\mathbf{X}}} J(\theta, \tilde{\mathbf{X}}, \mathbf{Y})) \tag{5}$$

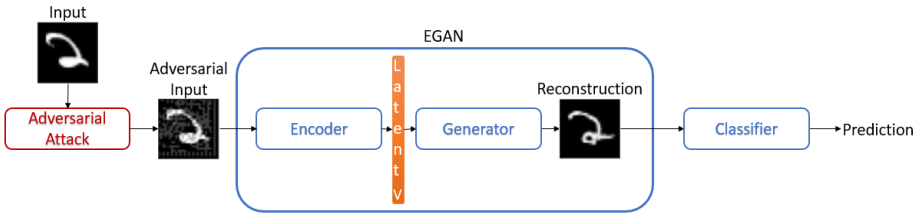

Fig. 1. *EGAN* methodology

where ϵ represents the step size in the direction that maximizes the loss, and $J(\theta, \tilde{\mathbf{X}}, \mathbf{Y})$ is the loss used to train the classifier. The higher the value of ϵ, stronger the attack and the easier it is to see with the naked eye.

At test time, a (possibly malformed) test image $\mathbf{X}$ is provided. Then the corresponding reconstructed image $\hat{\mathbf{X}}$ is computed by (3). Finally, the reconstructed image $\hat{\mathbf{X}}$ is passed on to a suitable image classifier.

Figure 1 shows a schema of the proposed method. First, given an adversarial input image, the trained encoder produces a latent vector from that input image. Then, the generator reconstructs the image from that latent vector. And finally, this reconstructed image is supplied as input to the classifier in order to predict its class.

The rationale behind our proposal is that the encoder learns to project an arbitrary input image $\mathbf{X}$ onto a latent vector $\mathbf{z}$ that belongs to the support of the probability distribution $P(\mathbf{z})$ of the latent vectors associated with the probability distribution of images $P(\mathbf{X})$ that was learned by the GAN. This way, if a malformed image from an adversarial attack is provided to the encoder, then the encoder projects the image onto a latent vector that is associated with a corrected image which belongs to the distribution of legitimate images learned by the GAN.

It must be highlighted that our proposed EGAN model is both class-agnostic and classifier-agnostic because the class labels are not used at any time, and there is no flow of information from the image classifier to the EGAN at all. In other words, the EGAN is a fully unsupervised neural model since the class label information is never employed, neither directly nor indirectly. This enhances the robustness of the EGAN as a defense against image classification attacks.

3 Experimental results

The experiment consists in compare different defense methods against a FGSM adversarial attack, measuring the accuracy of the defense methods across the different values of ϵ (input variation) and the computational time used to complete the experiment by applying these methods as preprocessors.

3.1 Methods

The methods used in the comparison are:

- *Original*: No defense method used.
- *DnCNN* [18]: Convolutional Neural Network trained to predict the noise of a certain sample. It is used for denoising and super-resolution.
- *AutoEncoder* [19]: Convolutional Neural Networks trained to encode and decode an image, making the information pass through a bottleneck and learning the significant features.
- *APE-GAN* [20]: Generative Adversarial Network trained to receive an adversarial sample as input and generate a sample without the adversarial modification.
- *PixelDefend* [21]: Auto-regressive Convolutional Neural Network trained to predict the value of a pixel based on the previous pixels. This network is used to make small changes on the (possibly malformed) sample.
- *Defense-GAN* [16]: Generative Adversarial Network trained to learn the training samples distribution. Then various random latent vectors are generated and optimized to generate reconstructed samples.

3.2 Dataset

The dataset used is the MNIST database (Modified National Institute of Standards and Technology database) which is formed by handwritten digits images and is divided into 60000 training images and 10000 testing images. It was created from another dataset called NIST (National Institute of Standards and Technology), where the training images and the testing images had a different origin. The MNIST was created mixing these images, anti-aliasing and resizing these images to 28x28 pixels.

3.3 Architecture and parameter selection

In our experiments, input images of size $28 \times 28 \times 1$ are considered, i.e., $N = 28$, $M = 28$, $Q = 1$; while the ϵ values used in the FGSM attack are from 0 to 1 where 0 means not modifying the input image and 1 means modifying the input image completely.

Regarding the proposed architecture of the encoder E, it is based on the GAN architecture called DCGAN [22], which introduces the use of convolutional layers instead of fully connected layers, in both generator and discriminator networks. This way, the encoder E is composed of four parts. The first part is a convolutional block that comprises a 2D convolutional layer with 3×3 kernel size that increases the number of channels to 100, followed by a batch normalization layer, and a leaky ReLU layer. The second part contains 10 convolutional blocks, each of them with a 2D convolutional layer with 3×3 kernel size that keeps the image size at $28 \times 28 \times 100$, followed by a batch normalization layer, and a leaky ReLU layer. The third part contains 11 convolutional blocks, each of them with

a 2D convolutional layer with 3×3 kernel size, followed by a batch normalization layer, and a leaky ReLU layer. Each convolutional block of the third part reduces the image size by two pixels to a final size of $6 \times 6 \times 100$. Finally, the fourth part contains a flatten layer whose output is a vector of size 3600×1, followed by a fully connected linear layer that outputs the latent vector $\mathbf{z}$ of size 30×1. Therefore, the dimension of the latent space is $L = 30$.

According to the classifier used to perform the experiments, it is a convolutional neural network composed of 2 2D convolutional layers with 3×3 kernel size. The first layer increases the number of channels to 20 and uses a stride of 2 while the second layer reduces the number of channels to 10 and uses a stride of 3. The both of this layers are followed by a batch normalization and a ReLU layer. After this the size of the data is $4 \times 4 \times 10$. Then we use 2 linear layers (the first followed by a ReLU layer) with 50 and 10 neurons respectively. Finally, a Log Softmax layer is applied to return the probabilities associated with each class. This trained network classifies no attacked MNIST images with an effectiveness of 99%.

3.4 Results

From a qualitative point of view, our proposed approach *EGAN* reconstructs images affected by an FGSM adversarial attack, even for highest values of ϵ, as can be observed in Figure 2.

Regarding the reconstructed images computed by the selected methods for the comparison, Figure 3 summarizes the visual results for each class of the considered dataset. As it is reported, *EGAN*, *APE-GAN* and *Defense-GAN* offer a reconstructed image with practically no noise.

In order to compare quantitatively the performance of the selected methods, the considered measure has been the accuracy (also known as detection rate). This measure shows the percentage of hits of the system by providing values in the interval $[0, 1]$, where higher is better.

As it can be seen in Figure 4, most considered methods yield a high similar performance for lower values of Epsilon (ϵ in equation 5). However, our proposal *EGAN* is the best method for values of Epsilon higher than 0.25.

Moreover, without loss of generality due to the required computational time of each method is the same independently of the value of ϵ, Figure 5 shows the computational time against the accuracy performance for $\epsilon = 0.5$. As it can be observed, our proposal is much faster than methods like *Defense-GAN*, that have a similar accuracy. On the other hand, faster methods such as *APE-GAN* do not offer a good performance for higher values of ϵ. This way, the proposed approach *EGAN* offers a good balance between computational time and accuracy.

4 Conclusions

This work proposes a methodology to reconstruct images that have been modified by applying a Fast Gradient Signed Method (FGSM) adversarial attack.

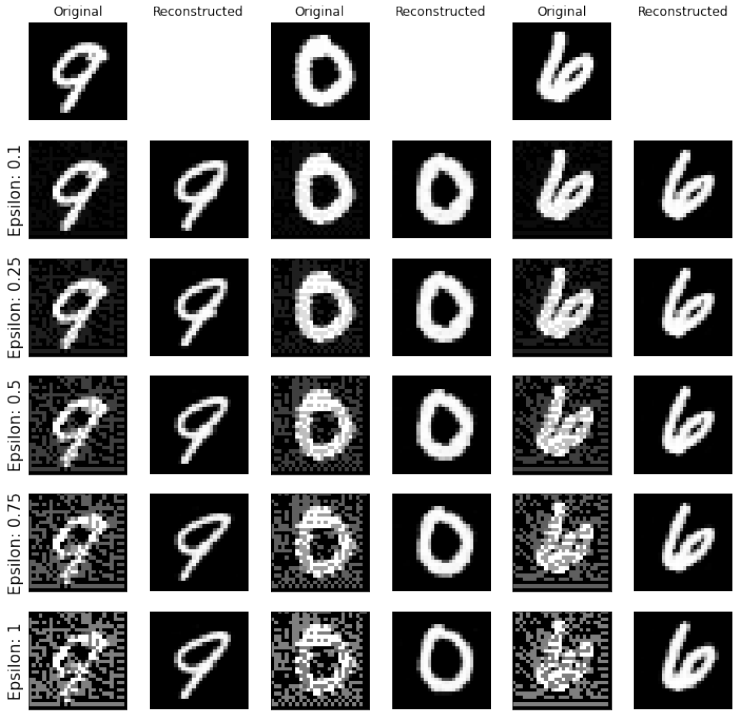

Fig. 2. Original, FGSM adversarial and reconstructed images with *EGAN*

This new approach is based on a Generative Adversarial Network (GAN) and an autoencoder. While the GAN produces an image from a latent vector, the encoder performs the inverse operation of the GAN by producing a latent vector from an image. Experiments by considering several well-known methods from the literature indicate that the performance of the proposed approach in terms of accuracy is suitable to face an adversarial attack. Additionally, the computational cost of the proposal is considerably more reduced than other methods of the same kind with similar yielded accuracy.

Acknowledgments

This work is partially supported by by the Ministry of Science, Innovation and Universities of Spain under grant number RTI2018-094645-B-I00, project name Automated detection with low cost hardware of unusual activities in video sequences. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name Anomalous behaviour agent detection by deep learning in low cost video surveillance intelligent systems. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer

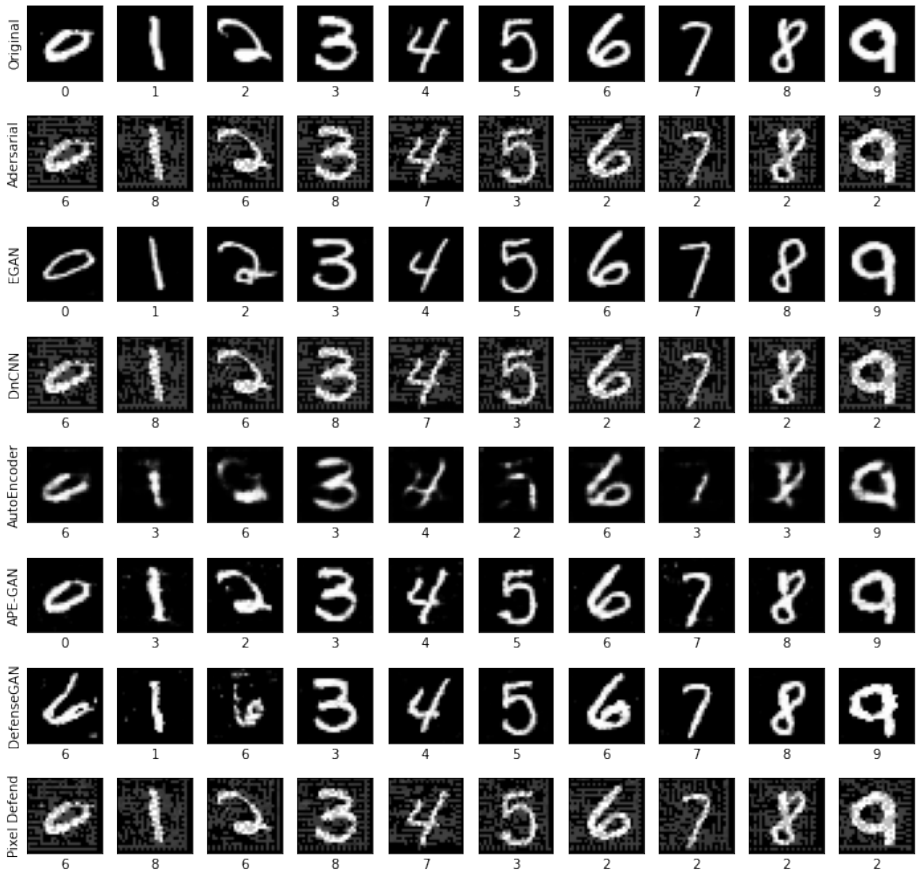

Fig. 3. Reconstruction comparison of FGSM adversarial attack with $\epsilon = 0.5$

resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. The authors acknowledge the funding from the Instituto de Investigación Biomédica de Málaga – IBIMA and the Universidad de Málaga.

References

1. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
2. G. Sivanarayana, K. N. Kumar, Y. Srinivas, and G. R. Kumar, "Review on the Methodologies for Image Segmentation Based on CNN," in *Communication Software and Networks*. Springer, 2021, pp. 165–175.

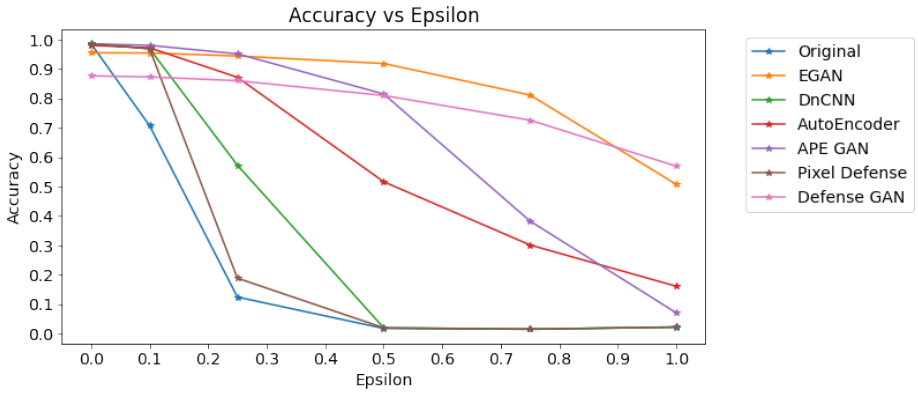

Fig. 4. Average performance of all considered methods. Note that the values of each method are connected together with lines to better compare the results, but this does not mean that the results are related.

3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
4. C. R. Steffens, L. R. Messias, P. J. Drews-Jr, and S. S. d. C. Botelho, "CNN Based Image Restoration," *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2020.
5. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.
6. Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.
7. Y. Tashiro, Y. Song, and S. Ermon, "Diversity can be transferred: Output diversification for white-and black-box attacks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
8. N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint arXiv:1602.02697*, vol. 1, no. 2, p. 3, 2016.
9. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
10. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
11. H. Lee, S. Han, and J. Lee, "Generative adversarial trainer: Defense to adversarial perturbations with gan," *arXiv preprint arXiv:1705.03387*, 2017.
12. C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.
13. X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, "Towards robust neural networks via random self-ensemble," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 369–385.
14. G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*, 2018.

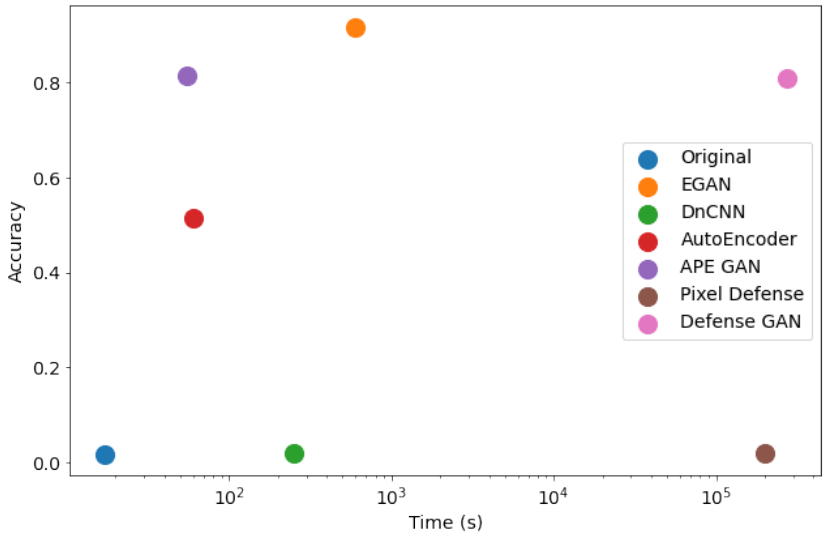

Fig. 5. Accuracy and computational time of different adversarial methods for $\epsilon = 0.5$

15. W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.
16. P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=BkJ3ibb0-
17. D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.
18. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
19. W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 490–497.
20. G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, "APE-GAN: Adversarial Perturbation Elimination with GAN," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3842–3846.
21. Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJUYGxbCW
22. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.